Introduction to Computer Science Application

Final Exam

Ugo Dal Lago* Marco Di Felice†

February 6, 2009

Exercise 1.
Given a function `fun`, defined as follows:

```
def fun(a,b):
    if (a>4):
        return 2*fun(b-1,a+1)
    elif (b<=5):
        return 2+ fun(a-1,b+1)
    else:
        return -1
```

Decide what is the value of the following three expressions: `fun(5,4)`, `fun(3,3)` and `fun(5,6)`

Solution.
The output is, respectively:
(i) `-2`
(ii) `5`
(iii) In this case, the function never ends.

Exercise 2.
What is the output of the following program?

```
def fun(d):
    return d.get('c',[0])
list=[]
dict={'a':[0,1],'b':[2],'c':[3,4]}
dict2=dict.copy()
dict3=dict
del dict2['c']
list=fun(dict3)
list=list + fun(dict2)
dict['a']=list
for k in dict3.keys():
    print k," ",dict3[k]
```

Solution.
The output is:

```
a   [3, 4, 0]
b   [2]
c   [3, 4]
```

Exercise 3.
What is the output of the following program?

```
import copy
class Point:
    pass
def createPoint(x,y):
    p=Point()
    p.x=x
    p.y=y
    return p
def sum(p):
    return p.x + p.y
def fun(p1,p2):
    p1.x=p2.y
    p2.x=p1.y
p1=createPoint(5,5)
p2=createPoint(0,0)
p3=p2
fun(copy.copy(p1),p3)
p3.y=10
print sum(p1) + sum(p2) + sum(p3)
```

Solution.
The output is `40`.

Exercise 4.
Write a function `replicateElements` which takes two arguments, a list `l` and a natural number `n` and returns another list obtained by replicating `n` times every element of `l`. For example `replicateElements([1,7,4],2)` should return `[1,1,7,7,4,4]`.
Solution.
A possible solution could be

```
def replicateElements(l,n):
    result=[]
    for el in l:
        result=result+[el]*n
    return result
```

Exercise 5.
A teacher keeps track of grades in a registry, which can be modeled by a dictionary `grades`. If a student's name is a string `a`, then `grades[a]` is a list of floating point numbers between 0

*dallago@cs.unibo.it
†difelice@cs.unibo.it

and 10, representing the grades obtained by `a`. Write

(i) A function `InitReg(list)` which returns a registry with no grades in it, for the students in `list`, a list of students' names.

(ii) A function `Average(grades,name)` which returns the arithmetical average grade for student `name`.

(iii) A function `Exam(grades)` which updates `grades` with grades obtained in an exam which has not been registered yet, assuming all students have participated to the exam. The grade obtained by a student should be obtained from the keyboard.

Solution.
Possible solutions could be

```
def InitReg(list):
    dict={}
    for el in list:
        dict[el]=[]
    return dict
def Average(grades,name):
    sum=0;
    list=grades[name]
    for el in list:
        sum=sum+el
    return sum/len(list)
def Exam(grades):
    for el in grades.keys():
        grade=float(input("What's the grade for "+el))
        grades[el]=grades[el]+[grade]
```

Exercise 6.
Write a function `flatten` which returns the flattening of an arbitrary nested list of natural numbers. For example, `flatten([[2,6,0],[[1],12],7])` should return the list `[2,6,0,1,12,7]`. *Hint:* you can use the built-in function `isinstance(o,t)` which returns `True` if and only if `o` is an object of type `t`.
Solution.
A possible solution could be

```
def flatten(a):
    if isinstance(a,int):
        return [a]
    elif len(a)==0:
        return []
    else:
        return flatten(a[0])+flatten(a[1:])
```